

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Carol Ann Egan et al.

Serial No.: 10/665,656

Filed: September 18, 2003

Group Art Unit: 2192

Confirmation No.: 8562

For: COMPUTER APPLICATION AND METHODS FOR
AUTONOMIC UPGRADE MAINTENANCE OF
COMPUTER HARDWARE, OPERATING SYSTEMS AND
APPLICATION SOFTWARE

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

CERTIFICATE OF ELECTRONIC TRANSMISSION

I hereby certify that this correspondence and any
enclosures are being electronically transmitted via
EFS-WEB on the date indicated below.

March 12, 2008

Date



Debra A. Peterson

**APPEAL BRIEF IN SUPPORT OF APPEAL
FROM THE PRIMARY EXAMINER TO THE BOARD OF APPEALS**

This is an appeal of a Final Rejection of claims 1-34 and 36-54 of Application
Serial Number 10/665,656 filed September 18, 2003. This brief is being submitted
pursuant to 37 C.F.R. 1.192. A Notice of Appeal was filed on January 14, 2008.

Docket No.: ROC920030111US1
Serial No.: 10/665,656

1. Real Party in Interest

International Business Machines Corporation is the real party in interest.

2. Related Appeals and Interferences

There are no related appeals or interferences pending with this application.

3. Status of Claims

Appellants appeal from the rejection in the November 14, 2007 Office Action of claims 1-34 and 36-54. The claims on appeal are set forth in Appendix A.

4. Status of Amendments

No amendments were filed subsequent to the final rejection of November 14, 2007.

5. Summary of Claimed Subject Matter

As described in the specification at page 3, paragraph 7, the present invention provides methods and a computer-readable program for providing autonomic, event driven upgrade maintenance of one or more software modules residing on a computer system. In a preferred embodiment, a method begins by detecting a predefined triggering event on the computer system indicative of a potential maintenance issue. Next, the computer system connects to an upgrade management server, where the upgrade maintenance server creates a list of recommended upgrade modules based upon the triggering event and a set of selection policies to download to the computer system. The

method then downloads the list of recommended upgrade modules from the upgrade management server to the computer system, and selectively installs upgrade modules chosen from the list of upgrade modules on the computer system. The user is then notified of the status of the upgrade maintenance operation.

Appellants are appealing from the Examiner's rejection of claims 1-34 and 36-54. Claim 1 is an independent claim. Claims 2-8 depend directly from claim 1. Claims 9-15 depend directly from claim 8. Claims 16-19 depend directly from claim 1. Claims 20-27 depend directly from claim 19. Claims 28-29 depend directly from claim 1. Claim 30 is an independent claim. Claims 31-34 and 36-37 depend directly from claim 30. Claims 38-44 depend directly from claim 37. Claim 45 depends directly from claim 30. Claims 46-53 depend directly from claim 45. Claim 54 is an independent claim.

In compliance with 37 C.F.R. § 41.37c(1)(v), a concise explanation of the subject matter defined in independent claims 1, 30 and 54, including references to the specification by page and line number, and to the drawings follow.

Claim 1 describes a method for providing autonomic, event-driven upgrade maintenance of one or more software modules residing on a computer system (Specification, page 3, paragraph 7, lines 1-3, and Fig. 4). The method begins by detecting a predefined triggering event on the computer system indicative of a potential maintenance issue, the predefined triggering event being triggered by a current operating condition of the computer system (Specification, page 14, lines 1-3, page 7, paragraph 20, and Fig. 4, element 304). Next, the method connects to an upgrade management server (Specification, page 14, paragraph 34, lines 3-5 and Fig. 4, element 306). The method then creates a list of recommended upgrade modules to download to the computer system on the upgrade management server, wherein the list is based upon a set of selection

policies (Specification, page 14, paragraph 34, lines 5-7 and Fig. 4, element 308). Next, the method downloads the list of recommended upgrade modules from the upgrade management server to the computer system (Specification, page 14, lines 7-10 and Fig. 4, element 310). Finally, the method selectively installs upgrade modules chosen from the list of recommended upgrade modules on the computer system (Specification, page 14, paragraph 34, lines 10-11 and Fig. 4, element 312).

Claim 30 provides a computer-readable program stored on a computer-readable medium (Specification, page 8, paragraph 24, and Fig. 4). The program begins by detecting a predefined triggering event on the computer system indicative of a potential maintenance issue, the predefined triggering event being triggered by a current operating condition of the computer system (Specification, page 14, lines 1-3, page 7, paragraph 20, and Fig. 4, element 304). Next, the program connects to an upgrade management server (Specification, page 14, paragraph 34, lines 3-5 and Fig. 4, element 306). The program then creates a list of recommended upgrade modules to download to the computer system on the upgrade management server, wherein the list is based upon a set of selection policies (Specification, page 14, paragraph 34, lines 5-7 and Fig. 4, element 308). Next, the program downloads the list of recommended upgrade modules from the upgrade management server to the computer system (Specification, page 14, lines 7-10 and Fig. 4, element 310). Finally, the program selectively installs upgrade modules chosen from the list of recommended upgrade modules on the computer system (Specification, page 14, paragraph 34, lines 10-11 and Fig. 4, element 312).

Claim 54 provides a method for deploying computing infrastructure, comprising integrating computer-readable code into a computing system, wherein the code in combination with the computing system is capable of providing autonomic, event-driven upgrade maintenance of one or more software modules residing on a computer system

(Specification, page 8, paragraph 24, and Figs 1 and 4). The method begins by detecting a predefined triggering event on the computer system indicative of a potential maintenance issue, the predefined triggering event being triggered by a current operating condition of the computer system (Specification, page 14, lines 1-3, page 7, paragraph 20, and Fig. 4, element 304). Next, the method connects to an upgrade management server (Specification, page 14, paragraph 34, lines 3-5 and Fig. 4, element 306). The method then creates a list of recommended upgrade modules to download to the computer system on the upgrade management server, wherein the list is based upon a set of selection policies (Specification, page 14, paragraph 34, lines 5-7 and Fig. 4, element 308). Next, the method downloads the list of recommended upgrade modules from the upgrade management server to the computer system (Specification, page 14, lines 7-10 and Fig. 4, element 310). Finally, the method selectively installs upgrade modules chosen from the list of recommended upgrade modules on the computer system (Specification, page 14, paragraph 34, lines 10-11 and Fig. 4, element 312).

6. Grounds of Rejection to be Reviewed on Appeal

The Examiner has rejected claims 1-54 under 35 U.S.C. § 102(a) as being anticipated by U.S. Patent 6,457,706 B1 (hereinafter Cheng). The issue is whether the Examiner is correct in asserting that claims 1-34 and 35-54 are anticipated by the Cheng reference.

7. Argument

Rejection under 35 U.S.C. § 102(a)

Appellants submit that claims 1, 30, and 54 require, in first element of each claim, **the detection of a triggering event on the computer system, wherein the predefined triggering event is triggered by a current operating condition within the computer system.** Support for this element can be found in the Specification of the present invention on page 7, paragraph [0020] which states as follows:

User and vendor supplied system monitors 116A, 116B on computer systems 102A, 102B interface with upgrade maintenance application 115C, 115D to monitor the operating state of the computer systems. A number of potential operating conditions may be monitored, including but not limited to: hardware configuration changes and new installations, software configuration changes and new installations, the number of errors detected, performance triggers, etc. Based on changes in the operating conditions of computer system 102A, 102B, a triggering event is generated, which is then detected by system monitor 116A, 116B...”

In contrast to the present invention (which relies on a **current operating condition** of the client computer system to initiate the trigger), Cheng et al. relies on: 1) **the user** to initiate the trigger; 2) a **time based periodic scheme** to initiate the trigger; or 3) **the service provider computer** to initiate the trigger on the client computer. Cheng et al., at column 7, lines 5-11 states as follows:

The update process 22 is typically initiated on the client computer 101. The user may manually initiate the process, or it may occur automatically, for example at preset periods, such as once a month. Alternatively, the process may be initiated by the service provider computer 102 prompting the client computer 101 at various intervals, or in response to particular events.

Appellants respectfully disagree with the Examiner's assertion that Cheng et al.'s time-based periodic event triggers (such as performing an activity once a month) are the same as events triggered by a current operating condition within the computer, as defined on page 7, paragraph 20 of the present invention (e.g., "hardware configuration changes and new installations, software configuration changes and new installations, the number of errors detected, performance triggers, etc."). The time based periodic scheme defined by Cheng et al. contemplates periodic (i.e., "synchronous") triggers that are initiated exclusively on a time basis and totally ignore the operating condition of the computer when generating the triggers. In contrast, the triggering events of the present invention rely on operating conditions that occur "asynchronously" within the computer (such as reaching an error threshold within the computer to generate the trigger).

For these reasons, Appellants submit that the Cheng et al reference does not anticipate element 1 of claims 1, 30, and 54 of the present invention.

Appellants further submit that the Cheng et al. reference does not anticipate element 3 of claims 1, 30 and 54 of the present invention, which states that the upgrade management server (i.e., known as the "service provider computer", Figure 1, element 102, in Cheng et al.) creates a list of recommended upgrade modules to download to the user/client computer system.

As stated on page 14, paragraph 34, lines 5-7 of the present invention, "At block 308, a list of recommended upgrade modules is created on the upgrade management server, based upon the triggering event and a set of selection policies." In contrast to the present invention, the client computer (not the service provider computer) in Cheng et al. makes the determination which software updates are applicable to relevant to the user's computer. As stated in Cheng et al., column 3, lines 33-35, "From client database, the client application determines which software updates are applicable or relevant to the

user's computer.” Thus, in the present invention, the list of recommended upgrade modules is done at the “server” level, not at the “client” level, as is shown in Cheng et al.

For this reason, Appellants submit that the Cheng et al. reference does not anticipate element 3 of claims 1, 30, and 54 of the present invention. Thus, Appellants submit that claims 1, 30, and 54 of the present invention are in condition for allowance.

Appellants further submit that dependent claims 2-29, 31-34, and 36-53, depend either directly or indirectly from claims 1 and 30, which for reasons stated above are now submitted as being in condition for allowance. Thus, claims 2-29, 31-34 and 36-53 are also now submitted as being allowable.

Also, with respect to claims 2 and 31, Appellants respectfully disagree with the Examiner's assertion that the step of notifying a user of the status of the upgrade maintenance operation is provided by Cheng et al. in Fig. 2, step 207. Appellants respectfully submit that the element from Cheng et al. specifically states “Display Information to Help User Select Updates”. In contrast to Cheng, the present invention relies solely on the “upgrade management server” (see claims 1 and 30, element 3) to generate the list of updates, and requires no user intervention. Thus, claims 2 and 31 of the present invention notify the user of what updates the upgrade management server has chosen, rather than querying the user to select the updates.

With respect to claims 3 and 32, Appellants respectfully disagree with the Examiner's assertion that the predefined triggering event comprises a change to the hardware configuration of the computer system is provided by Cheng et al. on page 4, paragraph 53. As stated previously, Cheng relies on either: 1) **the user** to initiate the trigger; 2) **a time based periodic scheme** to initiate the trigger; or 3) **the service**

provider computer to initiate the trigger on the client computer. A change to the hardware configuration of the computer system is not any of these, rather, it is triggered by a current operating event within the computer system, as claimed in element 1 of claims 1 and 30, from which claims 3 and 32 rely.

With respect to claims 4 and 33, Appellants respectfully disagree with the Examiner's assertion that the predefined triggering event comprises a change to the software configuration of the computer system is provided by Cheng et al. on page 4, paragraph 53. As stated previously, Cheng relies on either: 1) **the user** to initiate the trigger; 2) **a time based periodic scheme** to initiate the trigger; or 3) **the service provider computer** to initiate the trigger on the client computer. A change to the software configuration of the computer system is not any of these, rather, it is triggered by a current operating event within the computer system, as claimed in element 1 of claims 1 and 30, from which claims 4 and 33 rely.

With respect to claims 5 and 34, Appellants respectfully disagree with the Examiner's assertion that the predefined triggering event comprises exceeding a predefined error threshold within the computer system is provided by Cheng et al. on page 4, paragraph 53. As stated previously, Cheng relies on either: 1) **the user** to initiate the trigger; 2) **a time based periodic scheme** to initiate the trigger; or 3) **the service provider computer** to initiate the trigger on the client computer. Exceeding a predefined error threshold within the computer system is not any of these, rather, it is triggered by a current operating event within the computer system, as claimed in element 1 of claims 1 and 30, from which claims 5 and 34 rely.

With respect to claims 6 and 35, Appellants respectfully disagree with the Examiner's assertion that the predefined triggering event comprises exceeding a predefined performance threshold on the computer system is provided by Cheng et al. on

page 4, paragraph 53. As stated previously, Cheng relies on either: 1) **the user** to initiate the trigger; 2) **a time based periodic scheme** to initiate the trigger; or 3) **the service provider computer** to initiate the trigger on the client computer. Exceeding a predefined performance threshold on the computer system is not any of these, rather, it is triggered by a current operating event within the computer system, as claimed in element 1 of claims 1 and 30, from which claims 3 and 32 rely.

With respect to claims 7 and 36, Appellants respectfully disagree with the Examiner's assertion that the predefined triggering event comprises exceeding a predefined elapsed time since the last connection to the upgrade management server is provided by Cheng et al. on page 4, paragraph 53. As stated previously, Cheng relies on either: 1) **the user** to initiate the trigger; 2) **a time based periodic scheme** to initiate the trigger; or 3) **the service provider computer** to initiate the trigger on the client computer. Exceeding a predefined elapsed time since the last connection to the upgrade management server is not any of these, rather, it is an event triggered by a current operating event within the computer system, as claimed in element 1 of claims 1 and 30, from which claims 7 and 36 rely.

With respect to claims 8 and 37, Appellants respectfully disagree with the Examiner's assertion that the steps of connecting to an upgrade management server and selectively installing the list of recommended upgrade modules are controlled by a set of user defined policies is provided by Cheng et al. in Figure 2, item 207. Appellants submit that Figure 2, element 207 of Cheng et al. is a box on a flow diagram which is labeled as "Display Information to Help User Select Updates". In contrast to Cheng et al., the present invention automatically connects to the upgrade management server, and, after the list of recommended upgrades has been determined, selectively install this list without user intervention. As stated in the present invention on page 9, paragraph 27, lines 2-5,

“The user defined policies 172 include policies directed toward both connecting to the upgrade management server 104 and selectively installing a list of recommended upgrade modules returned by the upgrade management server 104 to the computer system 102.” Thus, while Cheng et al. requires the user to directly select the updates, the present invention utilizes the upgrade management server to automatically connect to the upgrade management server and selectively install the list of recommended upgrade modules without user intervention.

With respect to claims 9 and 38, Appellants respectfully disagree with the Examiner’s assertion that the set of user defined policies includes a preferred connection time is provided by Cheng et al. in paragraph 50, lines 2-7. As previously stated above, Appellants respectfully disagree that Cheng et al. provides “user defined policies”, but instead requires the user to manually select the modules for download (see Cheng et al. Figure 2, block 207). Thus, since Cheng et al. does not provide “user defined policies” in any context, it is a moot point that the set of user defined policies includes a preferred connection time.

With respect to claims 10 and 39, Appellants respectfully disagree with the Examiner’s assertion that the set of user defined policies includes the connection resource to be used is provided by Cheng et al. in paragraph 50, lines 2-7. As previously stated above, Appellants respectfully disagree that Cheng et al. provides “user defined policies”, but instead requires the user to manually select the modules for download (see Cheng et al. Figure 2, block 207). Thus, since Cheng et al. does not provide “user defined policies” in any context, it is a moot point that the set of user defined policies includes the connection resource to be used.

With respect to claims 11 and 40, Appellants respectfully disagree with the Examiner's assertion that the set of user defined policies includes the specification of computer system areas/software products to enable automatic application of upgrades is provided by Cheng et al. in paragraph 50, lines 2-7. As previously stated above, Appellants respectfully disagree that Cheng et al. provides "user defined policies", but instead requires the user to manually select the modules for download (see Cheng et al. Figure 2, block 207). Thus, since Cheng et al. does not provide "user defined policies" in any context, it is a moot point that the set of user defined policies includes the specification of computer system areas/software products to enable automatic application of upgrades.

With respect to claims 12 and 41, Appellants respectfully disagree with the Examiner's assertion that the set of user defined policies includes a defined time to connect to the upgrade management server to check for upgrades is provided by Cheng et al. in paragraph 50, lines 2-7. As previously stated above, Appellants respectfully disagree that Cheng et al. provides "user defined policies", but instead requires the user to manually select the modules for download (see Cheng et al. Figure 2, block 207). Thus, since Cheng et al. does not provide "user defined policies" in any context, it is a moot point that the set of user defined policies includes a defined time to connect to the upgrade management server to check for upgrades.

With respect to claims 13 and 42, Appellants respectfully disagree with the Examiner's assertion that the set of user defined policies includes a defined elapsed time interval for connecting to the upgrade management server to check for upgrades is provided by Cheng et al. in paragraph 50, lines 2-7. As previously stated above, Appellants respectfully disagree that Cheng et al. provides "user defined policies", but instead requires the user to manually select the modules for download (see Cheng et al.,

Figure 2, block 207). Thus, since Cheng et al. does not provide “user defined policies” in any context, it is a moot point that the set of user defined policies includes a defined elapsed time interval for connecting to the upgrade management server to check for upgrades.

With respect to claims 14 and 43, Appellants respectfully disagree with the Examiner’s assertion that the set of user defined policies includes a notification list for e-mailing user of information and actions relative to the upgrade management process is provided by Cheng et al. in paragraph 50, lines 2-7. As previously stated above, Appellants respectfully disagree that Cheng et al. provides “user defined policies”, but instead requires the user to manually select the modules for download (see Cheng et al. Figure 2, block 207). Thus, since Cheng et al. does not provide “user defined policies” in any context, it is a moot point that the set of user defined policies includes a notification list for e-mailing user of information and actions relative to the upgrade management process.

With respect to claims 15 and 44, Appellants respectfully disagree with the Examiner’s assertion that the set of user defined policies includes a list of one or more upgrade management servers to be used for the upgrade management process is provided by Cheng et al. in paragraph 50, lines 2-7. As previously stated above, Appellants respectfully disagree that Cheng et al. provides “user defined policies”, but instead requires the user to manually select the modules for download (see Cheng et al. Figure 2, block 207). Since Cheng et al. does not provide “user defined policies” in any context, it is a moot point that the set of user defined policies includes a list of one or more upgrade management servers to be used for the upgrade management process.

With respect to claims 19 and 45, Appellants respectfully disagree with the Examiner's assertion that the set of selection policies is sent from the computer system to the upgrade management server, citing Fig. 2 steps 208, 214 and related text. Appellants respectfully submit that Cheng et al. does not use selection policies in any context. Rather, Cheng et al. relies on the user to select the updates (see Fig. 2, step 207). The steps 208 and 214 cited by the Examiner actually occur after the user has already selected the updates (i.e., the user selects the updates at step 207), and merely download the updates and archive the changes.

With respect to claims 20 and 46, Appellants respectfully disagree with the Examiner's assertion that the set of selection policies includes creating the list of recommended upgrade modules based upon a specific set of upgrades requested by the computer system, citing Fig.2, step 206 and related text within Cheng et al. Appellants respectfully submit that Cheng et al. does not use selection policies in any context.

With respect to claims 21 and 47, Appellants respectfully disagree with the Examiner's assertion that the set of selection policies includes comparing the revision levels of the one or more software modules residing on the computer system against the revision levels of one or more software modules residing on the management server, citing Fig. 10, step 1006 and related text within Cheng et al. Appellants respectfully submit that Cheng et al. does not use selection policies in any context.

With respect to claims 22 and 48, Appellants respectfully disagree with the Examiner's assertion that the set of selection policies includes creating the list of recommended upgrade modules by identifying modules associated with a hardware change on the computer system, citing page 4, paragraph 54 within Cheng et al.

Appellants respectfully submit that Cheng et al. does not use selection policies in any context.

With respect to claims 23 and 49, Appellants respectfully disagree with the Examiner's assertion that the set of selection policies includes creating the list of recommended upgrade modules by identifying software modules associated with a software change on the computer system, citing page 4, paragraph 54 within Cheng et al. Appellants respectfully submit that Cheng et al. does not use selection policies in any context.

With respect to claims 24 and 50, Appellants respectfully disagree with the Examiner's assertion that the set of selection policies includes creating the list of recommended upgrade modules by identifying upgrades specifically associated with an error triggering event on the computer system, citing Fig. 2, step 206 and associated text within Cheng et al. Appellants respectfully submit that Cheng et al. does not use selection policies in any context.

With respect to claims 25 and 51, Appellants respectfully disagree with the Examiner's assertion that the set of selection policies includes creating the list of recommended upgrade modules by identifying upgrades specifically associated with a performance triggering event on the computer system, citing Fig. 2, step 206 and associated text within Cheng et al. Appellants respectfully submit that Cheng et al. does not use selection policies in any context.

With respect to claims 26 and 52, Appellants respectfully disagree with the Examiner's assertion that the set of selection policies includes creating the list of recommended upgrade modules by analyzing a problem history provided by the computer

system, citing Fig. 2, step 206 and associated text within Cheng et al. Appellants respectfully submit that Cheng et al. does not use selection policies in any context.

With respect to claims 27 and 53, Appellants respectfully disagree with the Examiner's assertion that the set of selection policies includes creating the list of recommended upgrade modules by identifying compatible revision levels between two or more software modules included within the list of modules, citing Fig. 2, step 205 and associated text within Cheng et al. Appellants respectfully submit that Cheng et al. does not use selection policies in any context.

8. Claims Appendix

1. A method for providing autonomic, event-driven upgrade maintenance of one or more software modules residing on a computer system, the method comprising:

detecting a predefined triggering event on the computer system indicative of a potential maintenance issue, the predefined triggering event being triggered by a current operating condition of the computer system;

connecting to an upgrade management server;

creating on the upgrade management server a list of recommended upgrade modules to download to the computer system, the list based upon a set of selection policies;

downloading the list of recommended upgrade modules from the upgrade management server to the computer system; and

selectively installing upgrade modules chosen from the list of recommended upgrade modules on the computer system.

2. The method of claim 1, wherein the method further comprises the step of:
notifying a user of the status of the upgrade maintenance operation.
3. The method of claim 1, wherein the predefined triggering event comprises a change to the hardware configuration of the computer system.

4. The method of claim 1, wherein the predefined triggering event comprises a change to the software configuration of the computer system.
5. The method of claim 1, wherein the predefined triggering event comprises exceeding a predefined error threshold on the computer system.
6. The method of claim 1, wherein the predefined triggering event comprises exceeding a predefined performance threshold on the computer system.
7. The method of claim 1, wherein the predefined triggering event comprises exceeding a predefined elapsed time since the last connection to the upgrade management server.
8. The method of claim 1, wherein the steps of connecting to a upgrade management server and selectively installing the list of recommended upgrade modules are controlled by a set of user defined policies.
9. The method of claim 8, wherein the set of user defined policies includes a preferred connection time.
10. The method of claim 8, wherein the set of user defined policies includes the connection resource to be used.
11. The method of claim 8, wherein the set of user defined policies includes the specification of computer system areas/software products to enable automatic application of upgrades.

12. The method of claim 8, wherein the set of user defined policies includes a defined time to connect to the upgrade management server to check for upgrades.
13. The method of claim 8, wherein the set of user defined policies includes a defined elapsed time interval for connecting to the upgrade management server to check for upgrades.
14. The method of claim 8, wherein the set of user defined policies includes a notification list for e-mailing user of information and actions relative to the upgrade management process.
15. The method of claim 8, wherein the set of user defined policies include a list of one or more upgrade management servers to be used for the upgrade management process.
16. The method of claim 1, wherein the one or more software modules comprises software applications.
17. The method of claim 1, wherein, the one or more software modules comprises operating systems.
18. The method of claim 1, wherein the one or more software modules comprises device drivers for installed hardware components.
19. The method of claim 1, wherein the set of selection policies is sent from the computer system to the upgrade management server.

20. The method of claim 19, wherein the set of selection policies includes creating the list of recommended upgrade modules based upon a specific set of upgrades requested by the computer system.

21. The method of claim 19, wherein the set of selection policies includes comparing a revision levels of the one or more software modules residing on the computer system against a revision level of one or more software modules residing on the upgrade management server.

22. The method of claim 19, wherein the set of selection policies includes creating the list of recommended upgrade modules by identifying modules associated with a hardware change on the computer system.

23. The method of claim 19, wherein the set of selection policies includes creating the list of recommended upgrade modules by identifying software modules associated with a software change on the computer system.

24. The method of claim 19, wherein the set of selection policies includes creating the list of recommended upgrade modules by identifying upgrades specifically associated with an error triggering event on the computer system.

25. The method of claim 19, wherein the set of selection policies includes creating the list of recommended upgrade modules by identifying upgrades specifically associated with a performance triggering event on the computer system.

26. The method of claim 19, wherein the set of selection policies includes creating the list of recommended upgrade modules by analyzing a problem history provided by the computer system.

27. The method of claim 19, wherein the set of selection policies includes creating the list of recommended upgrade modules by identifying compatible revision levels between two or more software modules included within the list of modules.

28. The method of claim 1, wherein the step of downloading the list of recommended upgrade modules from the upgrade management server to the computer system further comprises the step of downloading the upgrade modules themselves from the upgrade management server to the computer system.

29. The method of claim 1, wherein the step of selectively installing upgrade modules chosen from the list of recommended upgrade modules on the computer system further comprises the step of downloading any upgrade modules chosen from the list of recommended upgrade modules from the upgrade management server to the computer system prior to the install.

30. A computer-readable program stored on a computer-readable medium, said computer readable program being configured to perform the steps of:

detecting a predefined triggering event on a computer system indicative of a potential maintenance issue, the predefined triggering event being triggered by a current operating condition of the computer system;

connecting to an upgrade management server;

creating on the upgrade management server a list of recommended upgrade modules to download to the computer system, the list based upon a set of selection policies;

downloading the list of recommended upgrade modules from the upgrade management server to the computer system; and

selectively installing upgrade modules chosen from the list of recommended upgrade modules on the computer system.

31. The computer-readable program of claim 30, wherein the computer-readable program further includes the step of:

notifying a user of the status of the upgrade maintenance operation.

32. The computer-readable program of claim 30, wherein the predefined triggering event comprises a change to the hardware configuration of the computer system.

33. The computer-readable program of claim 30, wherein the predefined triggering event comprises a change to the hardware configuration of the computer system.

34. The computer-readable program of claim 30, wherein the predefined triggering event comprises exceeding a predefined error threshold on the computer system.

35. (Canceled)

36. The computer-readable program of claim 30, wherein the predefined triggering event comprises exceeding a predefined elapsed time since the last connection to the upgrade management server.
37. The computer-readable program of claim 30, wherein the steps of connecting to a upgrade management server and selectively installing the list of recommended upgrade modules are controlled by a set of user defined policies.
38. The computer-readable program of claim 37, wherein the set of user defined policies includes a preferred connection time.
39. The computer-readable program of claim 37, wherein the set of user defined policies includes the connection resource to be used.
40. The computer-readable program of claim 37, wherein the set of user defined policies includes the specification of computer system areas/software products to enable automatic application of upgrades.
41. The computer-readable program of claim 37, wherein the set of user defined policies includes a defined time to connect to the upgrade management server to check for upgrades.
42. The computer-readable program of claim 37, wherein the set of user defined policies includes a defined elapsed time interval for connecting to the upgrade management server to check for upgrades.

43. The computer-readable program of claim 37, wherein the set of user defined policies includes a notification list for e-mailing user of information and actions relative to the upgrade management process.

44. The computer-readable program of claim 37, wherein the set of user defined policies include a list of one or more upgrade management servers to be used for the upgrade management process.

45. The computer-readable program of claim 30, wherein the set of selection policies is sent from the computer system to the upgrade management server.

46. The computer-readable program of claim 45, wherein the set of selection policies includes creating the list of recommended upgrade modules based upon a specific set of upgrades requested by the computer system.

47. The computer-readable program of claim 45, wherein the set of selection policies includes comparing a revision levels of the one or more software modules residing on the computer system against a revision level of one or more software modules residing on the upgrade management server.

48. The computer-readable program of claim 45, wherein the set of selection policies includes creating the list of recommended upgrade modules by identifying modules associated with a hardware change on the computer system.

49. The computer-readable program of claim 45, wherein the set of selection policies includes creating the list of recommended upgrade modules by identifying software modules associated with a software change on the computer system.

50. The computer-readable program of claim 45, wherein the set of selection policies includes creating the list of recommended upgrade modules by identifying upgrades specifically associated with an error triggering event on the computer system.

51. The computer-readable program of claim 45, wherein the set of selection policies includes creating the list of recommended upgrade modules by identifying upgrades specifically associated with a performance triggering event on the computer system.

52. The computer-readable program of claim 45, wherein the set of selection policies includes creating the list of recommended upgrade modules by analyzing a problem history provided by the computer system.

53. The computer-readable program of claim 45, wherein the set of selection policies includes creating the list of recommended upgrade modules by identifying compatible revision levels between two or more software modules included within the list of modules.

54. A method for deploying computing infrastructure, comprising integrating computer-readable code into a computing system, wherein the code in combination with the computing system is capable of providing autonomic, event-driven upgrade maintenance of one or more software modules residing on a computer system, the method comprising the steps of:

detecting a predefined triggering event on the computer system indicative of a potential maintenance issue, the predefined triggering event being triggered by a current operating condition of the computer system;

connecting to an upgrade management server;

creating on the upgrade management server a list of recommended upgrade modules to download to the computer system, the list based upon a set of selection policies;

downloading the list of recommended upgrade modules from the upgrade management server to the computer system; and

selectively installing any upgrade modules chosen from the list of recommended upgrade modules on the computer system.

9. Evidence Appendix

There is no evidence attached for this appeal.

10. Related Proceedings Appendix

There are no related proceedings. Therefore, there are no copies of decisions rendered by a court of the Board attached here.

Appellants believe this appendix satisfies the requirements of 37 C.F.R. § 41.37(c)(x).

Respectfully submitted,

Date: March 12, 2008

By: 

James R. Nock
Registration No. 42,937
IBM Corporation - Department 917
3605 Highway 52 North
Rochester, Minnesota 55901-7829

Telephone: (507) 253-4661
Fax No.: (507) 253-2382